Best Practice Guide for Application Profiles and Ontologies

created in the context of the project

Developing interoperable metadata standards for contextualizing heterogeneous objects from collections, exemplified by objects of the provenance von Asch (ASCH)

funded by Deutsche Forschungsgemeinschaft (DFG)

http://asch.wiki.gwdg.de/

Revision history

Rev.	Date	Author	Organization	Description
0.1	09.04.2017	Susanne Al-Eryani	SUB Göttingen	Draft
0.2	20.09.2017	Stefanie Rühle	SUB Göttingen	Revision
0.3	27.11.2017	Susanne Al-Eryani	SUB Göttingen	Extended Draft
0.4	31.01.2018	Susanne Al-Eryani	SUB Göttingen	First version
0.5	29.03.2018	Stefanie Rühle	SUB Göttingen	Revision
0.6	24.05.2018	Susanne Al-Eryani	SUB Göttingen	Final Version

Table of Contents

1. Introduction

Ontologies are ten a penny nowadays and the underlying concepts are as diverse as the knowledge fields they represent. Before designing an ontology, it is therefore advantageous to enquire what is already there. For most knowledge domains, ontologies already exist and are used in one or more contexts. In most cases, to use one of these ontologies seems appropriate, because the design and maintenance of an ontology is a time-consuming and costly work, and the reuse of ontologies is not only cheaper but also supports the interoperability of a system with other systems of the same domain.[1]

Unfortunately, experience has shown that the available ontologies often only partly meet the requirements of particular projects ranging from minor differences to the incompatibility of relevant parts. In addition, new knowledge fields arise and the cross-domain description of knowledge fields becomes more and more important. Such shifting and changing of domains may require a new way of modelling the relations between things in a certain domain. In these cases, designing a new ontology may be inevitable.

This guide describes how to create an ontology and how to design an application profile based on this ontology. The guide gives issues to be considered, and steps and measures that must be taken. It takes into account questions about reusability and interoperability and describes techniques for documentation and publication. Chapter 2 defines the concepts and terms used in this guide. Based on the „Singapore Framework for Dublin Core Application Profiles“, Chapter 3 describes the preliminary research that is necessary and Chapter 4 illustrates the procedure of the design. Chapter 5 gives an overview of some metadata standards and Chapter 6 describes the selecting of terms for the application profile. Finally, Chapter 7 will be a short introduction to the design of a schema.

2. Concepts and Definitions

Due to the technological progress, the world is increasingly connected. The World Wide Web forms a dense network of information and has become an indispensable tool for the functionality of the digitized world. Evolving from the Web 2.0, where information is available via hyperlinks, Web 3.0 is targeting on the disposition of semantic embedment of data. This feature results from the interconnectedness of linkable data, known as Linked Data. Metadata is in this world of crucial importance, because it provides the basic structure for the interlinking of information in a Linked Data World. Metadata standards support the interoperability of the data, and ontologies and application profiles support its re-usability. In this chapter, we describe the main concepts relevant for the design of an ontology compliant to linked data applications. Our definitions may

---

1 https://www.researchgate.net/publication/305400728_Towards_a_Standard_Ontology_Metadata_Model

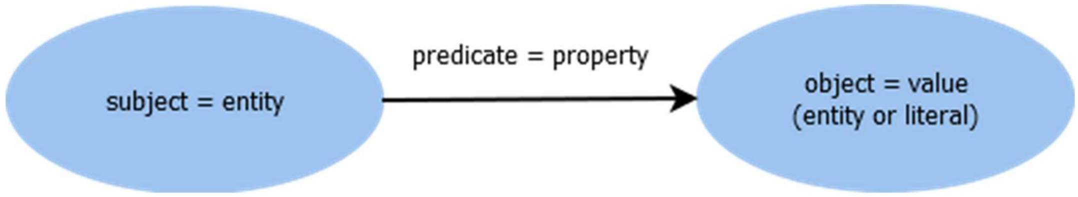
differ in some way from the use of these terms in other contexts and are an attempt to disambiguate terms often used with some semantic overlap.

2.1 Linked (Open) Data

Linked Data is a technique to publish metadata in the World Wide Web in a way that makes a semantic interlinking of the data possible. By using an open license for the data it becomes Linked Open Data and is free available and reusable. The key requirements of Linked Data where first defined by Tim Berners Lee in his Linked Data principles:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
- Include links to other URIs, so that they can discover more things.[2]

RDF (Resource Description Framework) is a format using URIs to make logical statements about resources. Applying the RDF model, statements are formulated as sentences consisting of triples with the components subject, predicate, and object. The relation between subject and object is expressed as a directed graph with the node "subject" as initial point, while the arc "predicate" defines its relation to the node "object".[3] Subjects are always entities and predicates are properties, identified by HTTP-URIs, but an object can be an entity identified by an HTTP-URI (used with an object property) or a literal statement (used with a datatype property), i.e. a combination of characters (letters, numbers, and symbols).



2.2 Metadata

Metadata is machine-readable structured data for the consistent description of entities. These entities can be material things (a physical object, e.g. a letter, a stone), digital objects (e.g. a digital representation of a material resource, a website), or non-physical items (e.g. an event, or a concept in an authority file). The data is structured, because it uses different attributes for the description of the different properties of an entity (e.g. for the name of a book we use the attribute "title", for the size of a book we use the attribute "extent"). It is consistent, because it uses the same attribute for the description of the same property of many entities (e.g. the use of the attribute "title" for

[2] https://www.w3.org/DesignIssues/LinkedData.html

[3] http://www.w3.org/RDF

all entities having a name). If these attributes are defined unambiguously, the data is machine-readable.

2.3 Metadata Model

A metadata model is a documentation specifying the design of a structured description of entities. It defines:

- what entities are described,
- how they are described,
  - which properties are relevant,
  - which terms are used for these properties,
- what relations exist between the entities,
- what terms are used to describe relations?

A model may depend on the used software or syntax (e.g., MODS or LIDO is used with XML) or it may be software and syntax independent (e.g. Dublin Core Terms may be used with XML, Turtle or others).

2.4 Metadata Standard

A metadata standard is a metadata model maintained and published by a standard setting body. These bodies may be part of a national or international standards organization (e.g. NISO), a domain specific association (e.g. the International Council of Museums) or a public body (e.g. the German National Library). Purpose and scope of a metadata standard is to maximize the interoperability of metadata and to support its re-usability. That is why a metadata standard should be software independent and publicly available.

2.5 Ontology

Objects and facts have a distinct meaning and an individual position within different knowledge environments. All facts, i.e. entities and relations between entities, are forming a net of information or knowledge graph. A knowledge environment embracing all pertinent facts occurring therein is an ontology. Apart from that broad concept, and as used in this guide, an ontology in a narrower sense is a metadata model, in which parts of that puzzle of information are structured and designed for its understandability and processability by a machine. Thus, an ontology can be referred to as a representation of those entities, properties and relations relevant to describe a particular area of interest in the knowledge environment. Because of their nature as knowledge graphs ontologies form the structure of the Linked Data World.

In dependence on the semantic complexity of the area of interest, the complexity of an ontology can go from simple to multifaceted. To limit this complexity, the creators of an ontology should use the possibilities of interlinking with ontologies from adjoining or cross-domain topics. This way the ontology supports the re-usability of metadata in a Linked Data environment, and becomes part and beneficiary of the growing network of information. An ontology can become a metadata standard, if it is maintained and

published by a standard setting body (e.g. CIDO-CRM maintained by the International Council of Museums and published as ISO 21127:2006 in 2006).

2.6 Application Profile

Being an active part of a network of institutions does not mean to share exactly the same standards for the description of resources. Each institution will make its local decisions about how resources should be described, and locally designed instructions inform about the use of metadata terms and vocabularies. Such a local metadata usage guide may be an application profile.[4] The Singapore Framework for Dublin Core Application Profiles defines an application profile as the "tailoring of standards to specific applications"[5]. By describing the classes, properties and relations relevant for a particular application, an application profile is a schema, created to support the functionalities of this application. Thereby it can be a general model used to integrate heterogeneous data from different providers (e.g. the application profiles of the German Digital Library or Europeana), or a specific model describing some area of expertise (e.g. a database for photosynthetic organisms). To ensure the interoperability with other applications, the Dublin Core Metadata Initiative (DCMI) recommends the reuse of terms defined in commonly used standards. If these terms are not specific enough, the creation of sub-terms (i.e. sub-classes and sub-properties) is possible.[6] DCMI calls this method the mixing and matching of metadata schemas.[7]

An application profile may serve different functions. Parts of it may be the specification used by the developer of an application and other parts may be rules used by cataloguers to enter data into the system. If the application profile includes a machine-readable schema, it supports the validation of the data. This way it assists the entry of compliant data and helps other users of the data to find out what to expect. Taking into account these roles of an application profile, the Singapore Framework distinguishes the following components of an application profile:

- the Functional Requirements, describing "the functions that the application profile is designed to support",
- the Domain Model, defining "the basic entities described by the application profile and their fundamental relationships",
- the Description Set Profile, specifying terms and constraints used for the description of the entities,
- the Usage Guidelines, describing "how the used properties are intended to be used",

[4] Miller, Steven J. (2011): *Metadata for Digital Collections: a How-to-do-it Manual.* New York, Neal-Schuman Publ.; p. 251.

[5] http://dublincore.org/documents/singapore-framework/

[6] Cf. the DCMI Dumb-Down Principals at https://github.com/dcmi/repository/blob/master/mediawiki_wiki/Glossary/Dumb-Down_Principle.md

[7] http://www.ariadne.ac.uk/issue25/app-profiles.

- the Encoding Syntax Guidelines, describing the "application profile-specific syntaxes and/or syntax guidelines".[8]

2.7 Knowledge Organizations Systems (KOS)

Natural languages use names to identify things. However, the names used by people for the same thing may differ, because in a natural language a thing can have many names. Even among experts within one community, various names for the same thing can be in use (synonyms like individual, human being, or person for people). Furthermore, one name can label different things (homonyms like 'bay' as a recess in a shore of the sea, a horse, or a laurel tree). In the metadata world, this variety of natural languages causes confusion, inaccuracy, and redundancies. Knowledge Organization Systems (KOS) are a way to avoid this diversity of terms. A KOS is a collection of terms, predefined, compiled and authorized by professionals of a special domain and used as values for the consistent description of resources. In a KOS, synonyms and ambiguous terms are related to the authorized terms, and homonyms are distinguished. In addition to the clear definition of the term, today the unique identification with a URI and its availability in the web are as important, because this allows the semantic contextualization and cross-referencing with other terms and supports the interlinking of data in a Linked Data World.[9]

3. From Case Studies to Requirements

Before we start with the design of our ontology, we have to scrutinize the domain and scope of our model. Ontologies may be modelled for a mail-order firm, the administration of a medical institution, a cultural heritage collection etc. Each of these domains has its own reality composed of different actors, activities and aims. To design an ontology that fits our purpose we must know more about this reality. For that reason, the fundamental questions we have to solve first are:

- Who will use the data?
- How will the data be used?
- What is the intention of the use?

The answers to these questions form the basis for our next step: the definition of those requirements that are relevant for our model and that have to be met by the use of it. In this chapter, we describe the steps needed to identify the functional requirements.

The Singapore Framework lists the Functional Requirements as one of the mandatory components of an application profile,[10] because the requirements support the proof of appropriateness of an application and the model it is based upon. A recommended way to find out about the requirements is to define how users would use the metadata in an

---

[8] http://dublincore.org/documents/singapore-framework/

[9] Cf. Zeng, Marcia Lei and Mayr, Philipp: Knowledge Organization Systems (KOS) in the Semantic Web. - http://www.iskouk.org/sites/default/files/Zeng-and-Mayr-ISKOUK17-web_3.pdf

[10] http://dublincore.org/documents/singapore-framework/

application. To formulate such use cases we have to identify who the users are. The users and their aims by using metadata can be identified due to the actions they perform within a knowledge environment. These actions may be of great diversity or may be a small number of the same actions, depending on the questions the users want to be have answered by the application. To find out, what the users' needs are and to transfer these needs into requirements we recommend the following four steps:

1. Collect and compile case studies.
2. Analyze the different scenarios within the case studies.
3. Define the requirements related to these scenarios.
4. Apply requirements to use cases.

3.1 Collect and Compile Case Studies

Different ways and platforms can help to collect and compile case studies. The methods vary from techniques like card sorting and free-listing exercises to time-consuming studies as interviews, examination of related artifacts and observation of user activities.[11] Using the example of the ASCH project, we show some of the methods we used within the project.

The project *Developing interoperable metadata standards for contextualizing heterogeneous objects, exemplified by objects of the provenance von Asch* (short: ASCH), founded by the German Research Foundation (DFG), was initiated by the State and University Library Göttingen (SUB Göttingen), and the Institute of Social and Cultural Anthropology and the Ethnographic Collection of the Göttingen University. The aim of the project was the development of a model that supports the structured and consistent description of provenance information and allows the contextualization of provenance descriptions from different scientific domains. Basis for the development of the model was an analysis of users' behavior and needs. The following survey methods were used:

- interviews with specialists from the subject fields in question;

To get information about the users and their behavior the staff of the ASCH project interviewed specialists from different scientific fields (Anatomy, Archaeology, Cultural and Social Anthropology, Geosciences, History of Art, Zoology) that were in the focus of interest for the project's work. The selection of the scientific fields resulted from the items that had to be taken into account and that are part of collections housed by different university departments.

- workshop with participants involved or interested in the issues of the subject fields in question;

For the same reason, the ASCH staff organized a workshop in the form of a World Café. Participants were scientists from different disciplines (Anatomy, Archaeology, Computer

[11] An overview about evaluation methods is described in Tonkin, Emma: Multilayered Paper Prototyping for User Concept Modeling: Supporting the Development of Application Profiles (see http://dcpapers.dublincore.org/pubs/article/download/990/951).

Science, Librarianship, Cultural and Social Anthropology, Geology, Geosciences, History, History of Art, Medicine, Mineralogy, Musicology, Philology, and Zoology) that were experts in the field of provenance research as this topic had to be considered as a crucial point by developing the metadata model.

- examining the material;

Not only the behavior of the people but also the items they are interested in were analyzed. The properties of different items, hosted by the university collections (the Historic Printed Collections, Manuscripts and Rare Books; the Ethnographic Collection; the Skull Collection; the Historical Collections of the Geoscience Centre; the Coin Cabinet; the Art Collection; and the Museum of Zoology) where examined and metadata describing these items was evaluated.

- identifying the requirements of the relevant metadata aggregators.

Because the model shall support the reuse of data by other applications, the requirements of two cultural heritage aggregators, the Deutsche Digitale Bibliothek (DDB)[12] and Europeana[13], were surveyed.

The result of such surveys will be examples from practice describing various ways of (re)search questions, activities, and visions occurring in the studied environment. These results can be transformed into case studies. Case studies compile the various results of the survey into a manageable amount of stories. They are hypothetical stories about actors pursuing a goal by one or more actions. It is possible to reproduce a complete real story or to select and recombine parts of it with parts of other examples. In each case, all (living) persons appearing in the case studies should be anonymized.

Case Study 1

Xiaomen is working on a study on the geographical extension (P) of the Great Auk at the end of the 18th century (T). She knows that the Museum of Zoology has an example (I) in its bird collection. On the museum's website she finds a list of terms, categorizing the different species. Choosing "more information about birds", she gets a taxonomy used to classify the species (C) of the bird collection. Browsing through this taxonomy, she gets a detailed description of the Great Auk. In a cross-database search she narrows her search to specimen (C) collected (E) in different geographic areas (P) from 1850 – 1900 (T). The description of one of the specimen (I) includes a digital image (D) of a stuffed auk (I) and a link to the description of access conditions (R) for the image. Xiaomen wants to use the digital image for her publication and follows the link.

[12] https://pro.deutsche-digitale-bibliothek.de/teilnahmekriterien [accessed 2017/08/09]

[13] http://pro.europeana.eu/page/edm-documentation [accessed 2017/08/09]

Case Study 2
Fritz has to identify origin (P) and age (T) of the coins (I) the University (A) got from Baron von Asch (A). He uses the correspondence and recordings of von Asch as a primary resource for his studies. He is especially interested in a tetradrachm showing Apollo (I) that according to the letters from von Asch was coined (E) in Athens (P) and collected (E) at the Crimea (P). He doubts that the information about the origin (P) of the coin (I) is right and wants to find out whether there are similar pieces (I) in other collections and where they were coined (E). Therefore, he searches for other collections with coins that are similar (C) to his.

3.2 Analyze the different scenarios and entities within the case studies

Each case study consists of one or more scenarios. A scenario describes a specific use of the system by a user and can consist of one or more actions and goals. On an abstract level we have to identify the actor (the ASCH project distinguishes between user, editor and aggregator[14]), and every single action with the system that the actor conducts in order to reach his goal. We also extract the entities, relevant for the case study and define them on an abstract level.

Case Study 1
In this case study, we extracted the following **scenarios**:

- Searching for an item (I) a **user** uses a taxonomy to get all items (I) of the same class (C).
- A **user** wants to know from which places (P) these items (I) have been collected (E) in a known time span (T).
- A **user** wants to use a digital representation (D) of an item (I) and needs information about the reusability (R) of the digital representation.

These scenarios consist of the following **actions**:

- **Action 1:** Search for items (I) that are instances of the same concept (C).
- **Action 2:** Search for those items (I) that were collected (E) in a known time span (T).
- **Action 3a**: Show the places (P) these items (I) were collected (E) from.
- **Action 4:** Search for digital representations (D) of an item (I).
- **Action 5:** Show the access information (R) about a digital representation (D).

Case Study 2
In this case study, we extracted the following **scenarios**:

- A **user** is searching for items (I) of the same class (C), sent by a known agent (A) to another known Agent (A).
- A **user** is searching for collections that contain items (I) of the same class (C) and of similar material (C) and appearance (C).

[14] In the ASCH project, the following actors were defined: the user as someone who is using metadata for searching, finding and selecting resources; the editor as someone who is creating, editing, annotating or manipulating metadata descriptions; the aggregator as someone who is harvesting and merging metadata of different origin. In the examples used in this guide only the user matters.

- A **user** wants to know where (P) these items (I) were coined (E).

These scenarios consist of the following **actions**:

- **Action 1:** Search for items (I) that are instances of the same concept (C).
- **Action 6:** Show those items (I) of this class that are sent (E) by a known Agent (A) to a known Agent (A).
- **Action 7:** Search for those items (I) of this class (C) that have similar material (C) and appearance (C).
- **Action 3b:** Show the places (P) these items (I) were coined (E) at.

As we see, actions related to the case studies may differ but may also be the same in different case studies, e.g. Action 1 is relevant for both case studies.

3.3 Define the requirements

Each of the actions will meet one or more requirements necessary to complete, to ensure that the interaction between an actor and the metadata application is successful. The requirements are results-oriented and should be described generically, shortly and precisely.

Actions	**Requirements**
Action 1: Search for items (I) that are instances of the same concept (C).	**R1:** The nature of an item (I) must be described using concepts (C).
Action 2: Search for those items (I) that were collected (E) in a known time span (T).	**R2:** An item (I) must be related to events (E). **R3:** The nature of an event (E) must be described using concepts (C). **R4:** An event (E) must be related to the time (T) it happened.
Action 3a: Show the places (P) these items (I) were collected (E) from. **Action 3b:** Show the places (P) these items (I) were coined (E) at.	**R2:** An item (I) must be related to events (E). **R3:** The nature of an event (E) must be described using concepts (C). **R5:** An event (E) must be related to the place (P) it occurred.
Action 4: Search for digital representations (D) of an item (I).	**R6:** An item (I) should be related to digital representations (D).
Action 5: Show the access information (R) about a digital representation (D).	**R7:** The description of a digital representation (D) must provide information about the rights (R).
Action 6: Show those items (I) that are	**R2:** An item (I) must be related to

sent (E) by a known Agent (A) to a known Agent (A).	events (E). **R3:** The nature of an event (E) must be described using concepts (C). **R8:** An event (E) must be related to agents (A) involved in the event.
Action 7: Search for those items (I) that have a similar material (C) and appearance (C).	**R1:** The nature of an item (I) must be described using concepts (C).

3.4 Apply requirements to use cases

The use cases define the use of the metadata by the actors in general. They are the essence of the different scenarios, and cluster scenarios, actions and requirements on an abstract level thereby enabling a view of the requirements from a more general perspective, which is especially useful when we have to handle a high number of detailed requirements. We cluster our examples by the following use cases, defined by the ASCH project[15] and apply the above listed requirements:

Use Case 1: Scenarios related to the identification of resources.

- **R1:** The nature of an item (I) must be described using concepts (C).
- **R3:** The nature of an event (E) must be described using concepts (C).
- **R6:** An item (I) should be related to digital representations (D).

Use Case 2: Scenarios related to information about events/activities the items were involved in.

- **R2:** An item (I) must be related to events (E).
- **R4:** An event (E) must be related to the time (T) it happened.
- **R5:** An event (E) must be related to the place (P) it occurred.
- **R8:** An event (E) must be related to agents (A) involved in the event.

Use Case 3: Scenarios concerning the usability of resources.

- **R7:** The description of a digital representation (D) must provide information about the rights (R).

4. Entities, Properties and Relations

In addition to the examination of user interactions, an examination of the items we want to describe with the metadata is important. We need to know what kind of items will be described with the metadata and what data already exist about these items.[16] Talking to

[15] Use Case 1 is equal to “UC 2 Identification of resources”, Use Case 2 is equal to “UC 3 Information about the history/lifecycle of resources” and Use Case 3 is equal to “UC 7 Access to resources”, all defined by the ASCH project (see http://asch.wiki.gwdg.de/index.php/Use_Cases#Use_Cases).

[16] In the ASCH project, the analysis of items and metadata describing these items was part of the collection and compilation of case studies (see chapter 3.1.).

users and metadata editors, we can find out how satisfied they are with the data available and what is missing. The material we have to analyze may be very voluminous and heterogeneous, depending on the given context. It may consist of single items or convolutes of items, data about persons, organizations, places, events, etc. During the process, a reconciliation of the items, the available data and the requirements we defined is recommended. The result of the process should be a list of entities and their relations. Each of the entities may be associated with one or more other entities in different ways. Based on the requirements, we are able to identify which of these entities and relations are relevant for our domain model.

In this chapter, we describe the interaction of entities, properties and classes and show how to design a domain model on the base of the requirements.

4.1 Entities – Classes – Subclasses

> *„An entity is something that exists as itself, as a subject or as an object, actually or potentially, concretely or abstractly, physically or not."*[17]

In this guide, we use the term entity for all that we want to describe with our metadata, i.e. our understanding of the term complies with the definition of 'Resource' in RDF-Schema[18]. An ASCH entity is therefore always an instance of rdfs:Resource. In the ASCH project, some of these entities are the items, sent by Baron von Asch to the University of Göttingen.

During a long period, Baron Georg Thomas von Asch (1729 – 1807) sent a large number of artifacts, natural items and documents from Saint Petersburg to Göttingen, including fossils, manuscripts, maps, skulls, coins, medals, rocks, drawings, etc. These items were incorporated into the Royal Academic Museum and into the University Library. Later on, the items were dispersed and are now to be found in the various specific collections of the Göttingen University. In correspondence with the discipline-specific alignment of these institutions, we find not only different ways of how to describe an item, but also a wide array of models being used to do so.

To categorize the amount and diversity of the items Baron von Asch had collected and sent to the University and the entities related to them is beyond the scope of this guide. Therefore, we narrow it down to those entities referred to in our examples. We first categorize those on a general level by defining classes under which these entities will be grouped.

We will see that some of the classes (e.g. the class Digital Representation (D) and the class Rights (R)) are relevant for only one of the two Case Studies, but most of them can be found in both of them.

[17] https://en.wikipedia.org/wiki/Entity (last edited on 26 November 2017, at 13:56).

[18] „All things described by RDF are called resources, and are instances of the class rdfs:Resource. This is the class of everything. All other classes are subclasses of this class" (see: https://www.w3.org/TR/rdf-schema/#ch_resource).

Case study 1	Case study 2	Class
• geographical extension • different geographic areas	• origin • Athens • Crimea	Place (P)
• end of the 18th century • from 1850 - 1900	• age	Time (T)
• example of the Great Auk • one of the specimen • stuffed auk	• coin • tetradrachm showing Apollo • similar piece	Item (I)
• taxonomy used to classify the species • specimen		Concept (C)
• collected	• coined • collected	Event (E)
	• University • Baron von Asch	Agent (A)
• digital Image		Digital Representation (D)
• description of access conditions		Rights (R)

Now we define these classes according to the meaning they have in our context.

- Agent (A): A person, organization or group that acts or may act.
- Concept (C): Controlled values representing an idea or a unit of thought.
- Digital Representation (D): A digital resource used as a proxy for a real-world-thing in a digital environment.
- Event (E): Something that happened and affected the lifecycle of a resource.
- Item (I): A real world thing.
- Place (P): A geographic location.
- Rights (R): Information about copyright, usage and access rights of a digital object.

A class is a group of entities that have common properties.[19] The entities that are members of this class are called "instances" of this class[20] and an entity may be an instance of many classes, depending on the context in which it is used. We may define

[19]Cf. https://www.w3.org/TR/owl-ref/#Class (OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004).

[20] https://www.w3.org/TR/rdf-schema/#ch_classes

classes on a more general level, as we did in the list above, or on a more granular level if it seems appropriate. It is e.g. appropriate to assign such different things like the “Great Auk” and the “Tetradrachm showing Apollo” to the class “Item” as long as we want to describe their relation to provenance information, and the properties they have in this context are the same. If we want to describe properties that differ between entities, e.g. the depiction on a coin, we have to decide whether it is useful to define a subclass for those entities. A subclass is a class that groups only some of the instances of a class, which are those that have properties in common that the others do not have. Moreover, all properties that apply to the class, also apply to the subclasses. In the following example, we define subclasses for the classes “Item” and “Agent”, so that we may differentiate between more granular specifications of these classes. How granular the classes are needed, depends on our requirements.

Class	Subclass
Item: A real world thing.	
	• **Numismatic Item:** A thing used as money in the present, past or future (e.g. a coin or a banknote).
	• **Organism:** Natural item living or having lived as a human, animal, plant etc.
Agent: A person, organization or group that acts or may be act.	
	• **Person:** An individual human being.
	• **Corporate body:** A group of people recognized by law as one agent.

4.2 Properties and Relations

Now, knowing which entities will be described, we need to define what statements we want to make about these entities. As described above, the basic principle for the categorization of entities and classes are those properties that the entities have in common. These properties may be more general, like the name of the entity or the type, or more specific, like the extent, the color or, as in our example, the depiction on an entity. To see which of the properties are relevant for our example, we again look at our requirements and select the properties that match.

Requirement	Entity	Property	Entity
R1: The nature of an item (I) must be described using concepts (C).	Item	**has type**	Concept
R2: An item (I) must be related to events (E).	Item	**was affected by**	Event
R3: The nature of an event (E) must be described using concepts (C).	Event	**has type**	Concept

R4: An event (E) must be related to the time (T) it happened.	Event	**occurred at**	Time
R5: An event (E) must be related to the place (P) it occurred.	Event	**took place at**	Place
R8: An event must be related to agents (A) involved in the event.	Event	**has participant**	Agent
R6: An item (I) should be related to digital representations (D).	Item	**has representation**	Digital Representation
R7: The description of a digital representation (D) must provide information about the rights (R).	Digital Representation	**has rights**	Rights

The above listed properties are called object properties, because they define the relationship between two entities or – as owl calls it – "link individuals to individuals".[21] Relationships like these are the backbone of a machine-readable semantic contextualization. However, at some point of the description we need some human readable string, telling the user what it is he is looking at. Properties used with such strings are called datatype properties, because they "link individuals to data values". [22] Properties often used with strings are names, date information or summaries. The following table shows the use of object properties and datatype properties for our examples. An X represents the values of the datatype properties; for the values of the object properties, we use the acronyms of the classes.

Properties / Classes	has name	has type	was present at	took place at	occurred at	has participant	has representation
Item (I)	X	C	E				D
Event (E)	X	C		P	X	A	

If we compare this table with a table about one of our subclasses, the Numismatic Item, we see, that all properties used with Item may also be used with its subclass Numismatic Item, plus those properties that are only relevant for the Numismatic Item (in this example the properties has material and has depiction).

Properties / Classes	has name	has type	was present at	has representation	has material	has depiction
Item	X	C	E	D	C	C

[21] https://www.w3.org/TR/owl-ref/#Property (10.Feb. 2004)
[22] https://www.w3.org/TR/owl-ref/#Property (10. Feb. 2004)

Just as the entities, we define the properties according to the meaning they have in our context.

- has depiction: Relates to a controlled value describing a resource shown on an item.
- took place at: Relates to a place an event happened.
- has material: Relates to a resource describing a substance an item is made of.
- has name: A name or appellation that identifies the resource and allows the distinction between resources of the same type.
- has participant: Relates to an agent who participated in an event.
- has type: Relates to a controlled value describing the nature of the resource.
- was affected by: Relates to an event in the lifecycle of an item.
- occurred at: The date an event happened.
- has representation: Relates to a digital representation or copy of the item.

Here too, we have the possibility to define more granular terms – so called sub properties – and group them under properties that are more general. The property "has material" e.g. could be defined as a sub property of "has type". To do so, it is important to realize, that the property to which a sub property is related, may have constraints concerning the domain and range that must be met by the sub property. Domain and range specify the classes of which entities used with this property must be instances, whereas the domain defines the class of those entities that are described by the property, and the range the class of those entities that are values used with the property. We may e.g. set the rules, that our sub property "has material" may be used only, when the described resource is a "Numismatic Item", and that the value, used with this property must always be an instance of the class "Concept". These rules will work, as long as its super-property "has type" has no constraints about domain and range, or as long as its domain and range are specified with the same class or a superclass of it. E.g., in the following definition the relation between the property and the sub property only work because Numismatic Item is a subclass of Item.

Property	has type
Domain	Item
Range	Concept

Property	has material
Sub Property of	has type
Domain	Numismatic Item
Range	Concept

4.3 The Domain Model

Knowing the entities and their classes on the one hand and the properties, representing the relations between these entities on the other, we are now able to create the domain model, an abstract representation of the concepts relevant for our knowledge domain. In the Singapore Framework, the domain model is described as another mandatory component of an application profile, which defines the "basic entities described by the application profile and their fundamental relationships".[23] The Singapore Framework leaves developers free how to express the model, but we recommend using a more formal approach. The diagrams show the domain model of our example as a graph

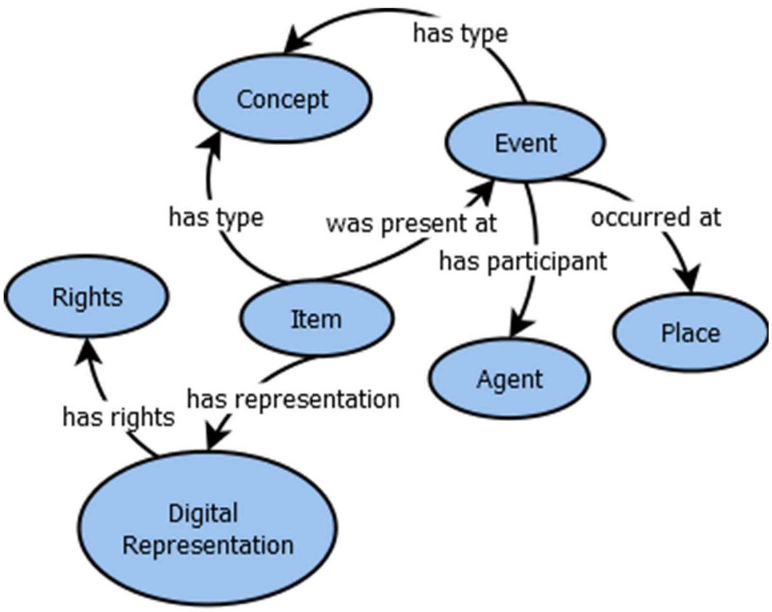


Figure 1: Domain Graph of the Model

representing the semantic contextualization of the different entities (see Figure 1) and as a UML diagram, describing not only the relations between the entities but also the

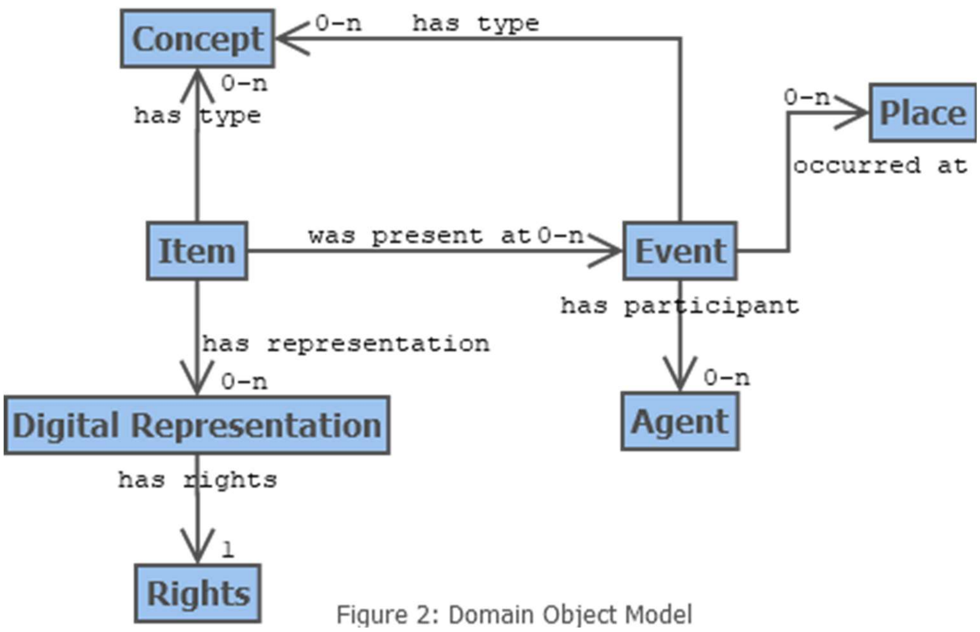


Figure 2: Domain Object Model

[23] http://dublincore.org/documents/singapore-framework/

occurrence of the properties used to describe these relations (see Figure 2). What approach we will finally choose will depend on the complexity of our model and the tools available.

5. Metadata Standards

In recent years, more and more organizations are eager to digitize their holdings and provide access for a broader audience on the web. Presenting those holdings in a form that would not only allow others to reuse and understand its entities but also to interlink them with other entities on the web necessitates metadata about these entities that is interoperable with data from other applications. However, where does metadata interoperability come from in a decentralized world where "everybody can say everything about everything"[24] the way he or she likes? The "Interoperability Levels for Dublin Core Metadata" published by the Dublin Core Metadata Initiative (DCMI) in 2009 are an attempt to give an answer.[25] Describing a "ladder of interoperability", DCMI defines four steps, where the first step is the use of common metadata terms that "provide a basis for sharing meanings within and between groups of people". These common terms is something we will find in metadata standards, usually designed and used by groups of experts from one or more domains. The decision about what metadata standard will be used depends on the context and the relevant stakeholders. Important is, that in addition to domain specific standards appropriate for a specific knowledge community, we should also keep an eye on more general standards. At least, the usage of or the interlinking with common and widely recognized standards increases the usefulness of our metadata and facilitates the integration of data into different contexts.

In this chapter, we present some of the linked data compliant metadata standards we examined in the ASCH project. We differentiate between metadata models for the description of entities, on the one hand, and knowledge organization systems (KOS) as sources for unique entity identifiers, on the other. We also differentiate between cross-domain standards widely used and accepted in the Linked Data World and domain specific standards, used by special knowledge communities (e.g. biologists). Due to the variety of ontologies created in the last years, we can only introduce to some vocabularies used in the Linked Data Cloud. For those who need a broader overview we recommend to visit directories like the Open Metadata Registry[26] or the Linked Open Vocabularies (LOV)[27]. Another lookup service, especially for KOS is BARTOC.org, the Basel register of thesauri, ontologies and classification.[28] If you want to know more

---

[24] Bizer, Chris: Introduction to Linked Data, September 12th, 2011, Leipzig, Germany (see http://www.ksi.mff.cuni.cz/~knap/NSWI144/2013/1/Seminar1.pdf)

[25] http://dublincore.org/documents/interoperability-levels/

[26] http://metadataregistry.org/

[27] http://lov.okfn.org/dataset/lov/

[28] https://bartoc.org/

about metadata standards in general, we recommend reading "Understanding Metadata", published by NISO in 2017.[29]

5.1 Cross-Domain Models

Dublin Core

At first developed for the description of web resources in 1995, the Dublin Core Metadata Element Set (DCMES)[30], also known as ISO 15836, has become a popular standard for the description of all kinds of resources. Its limited number of fifteen elements offers a very general level for resource description that is simple to understand and easy to implement. Furthermore, with the DCMI Metadata Terms[31], the Dublin Core Metadata Initiative (DCMI) has enhanced the standard by terms that are more granular and sub-terms that allow a more sophisticated description of entities and their relations in the Linked Data World. Today, the DCMES and the DCMI Metadata Terms are widely used by web standards (e.g. OAI-PMH or the Europeana Data Model).

Schema.org

Founded in 2011 by Google, Microsoft, Yahoo and Yandex, schema.org provides a markup language for the identification of web site content. The language is not bound to any distinct programming language and can be used with a variety of encodings, e.g. RDFa, Microdata and JSON-LD.[32] In collaboration with different communities, the schema has been expanded in the last years and the core vocabulary nowadays provides more than 1.500 Types, Properties and Enumeration Values grouped into ten sections (e.g. Creative works, Event, Health and medical types, etc.).[33]

Friend of a Friend (FOAF) Specification

Starting with its creation in mid-2000, the FOAF Ontology soon became an important element of the Social Semantic Web. Used to describe the relations between people and entities (like organizations, activities, webpages, etc.) related to these people, it supports the interlinking of people and information about them using W3C's RDF technology. The core set of FOAF offers 19 core terms, properties and classes, used in different context. Further terms defined in the Social Web Set allow a more granular description of persons and related resources.[34]

Simple Knowledge Organization System (SKOS)

Another member of the family of Semantic Web standards is SKOS.[35] Published in 2009 as a W3C recommendation, it is an ontology designed to describe Knowledge Organization Systems (KOS) like thesauri, classification schemes, taxonomies, subject

---

[29] Riley, Jenn: Understanding Metadata: What is Metadata, and What is it For?: A Primer. January 01, 2017 (http://www.niso.org/publications/understanding-metadata-2017).

[30] http://dublincore.org/documents/dces/

[31] http://dublincore.org/documents/dcmi-terms/

[32] http://schema.org/

[33] http://schema.org/docs/schemas.html

[34] http://xmlns.com/foaf/spec/

[35] https://www.w3.org/2004/02/skos/

heading systems, etc. SKOS supports the decentralized use of KOS as machine-readable vocabularies usable in a wide range of applications. Like most of the cross-domain metadata standards, the number of SKOS terms is limited to those essential for the purpose and is expandable for granular descriptions.[36]

Data Model for Provenance Interchange on the Web (PROV)

Another recommendation of the W3C is the Provenance Ontology (PROV-O)[37], a linked data compliant model for the description and exchange of provenance information. The development is focused on the description of the lifecycle of web documents, but it can be used for the description of all sorts of provenance, independent from the character of the entities described. The core set of PROV-O are the starting point classes and properties, a more granular description is possible by using the sets of expanded and qualified properties and classes. In co-operation with DCMI, the Provenance Working Group created a mapping from Dublin Core to PROV to support the use of Dublin Core data in a PROV environment.[38]

International Image Interoperability Framework (IIIF)

IIIF is a standard for the identification and description of images, their structure, layout, and related content. The standard is maintained by the IIIF Consortium, a community of cultural heritage institutions, universities and software firms,[39] and provides a technology that allows access to image-based resources hosted all over the world. IIIF provides a set of application programming interfaces (API) supporting the interoperability of images across viewer applications and repositories.[40] Using JSON-LD for the encoding of the data, IIIF ensures its semantic embedding, persistence, and reusability.[41]

5.2 Cross-Domain KOS

In the linked data world, the identification of entities using URIs is one of the premises behind the interlinking of information. Interlinking only works, however, if all data providers use the same URIs or align their URIs to the most often used. The entities identified by a URI must be defined by machine-understandable data in order to assure that the meaning of an entity is clear. In this section, we introduce into some cross-domain sources for URIs that are more or less widely used in the WWW.

Wikidata

Wikidata is a knowledge base, hosted by the Wikimedia Foundation, storing and providing structured data from Wikipedia, Wikivoyage, Wikisource, and other sources. All data is available under a free license and may be used by everyone. The content of the database is focused on entities, which in Wikidata are called items. Every item is

[36] https://www.w3.org/2009/08/skos-reference/skos.html
[37] https://www.w3.org/TR/prov-o/
[38] https://www.w3.org/TR/2013/NOTE-prov-dc-20130430/
[39] http://iiif.io/community/
[40] http://iiif.io/technical-details/
[41] http://iiif.io/api/annex/notes/jsonld/

identified by a unique URI, is described by consistent and structured metadata, and is interlinked with other items within Wikidata and other entities on the web.

Gemeinsame Normdatei (GND)

The GND is a German authority for persons, corporate bodies, concepts and other entities. The database is a collaborative effort of the German National Library (DNB) and the union catalogues of the German speaking countries and other institutions. The data is freely available and is mainly used by libraries in the German speaking countries and, indeed, more and more by archives, museums and in project contexts.[42] Just as the Wikidata items, the GND entities are identified by unique identifiers and are interlinked with entities inside and outside the GND.

ORCID

ORCID is an international nonprofit organization, funded through membership with the purpose to solve the problem of the name ambiguity of personal names by providing unique identifiers for people participating in “research, scholarship and innovation”, and by interlinking these people with their works or other available contributions.[43] It is a collaborative approach, wherein individuals have to register and maintain the data by themselves to get an ORCID identifier. The identifier and the data related to it may than be integrated in different research systems and workflows manually or via interfaces.

Virtual International Authority File (VIAF)

Another authority service for personal names is VIAF. Hosted by OCLC, this service matches name authority files from national libraries and other agencies, like the Getty Union List of Artist Names (ULAN) and Wikidata.[44] It provides URIs for all entities merged in this process. Because of its multilingualism, it is an important element in the development of Linked Data in Cultural Heritage Institutions.

GeoNames

GeoNames is an international multilingual geographical database. It matches data from different sources worldwide, mostly official public agencies. Via a wiki interface it allows a manually revision of the data by editors. The free download of the data is possible through some web services and a daily database export. URIs are used to identify the GeoNames entities and link them to DBpedia and other Linked Data Services.[45]

5.3 Domain specific models

CIDOC Conceptual Reference Model (CIDOC CRM)

CIDOC CRM is an ontology for the description of concepts and relationships of cultural heritage. It was developed by the CIDOC Documentation Standards Group in the International Committee for Documentation of the International Council of Museums (ICOM) and was published as ISO 21127:2006. Cultural heritage information, provided by

[42] http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html
[43] https://orcid.org/
[44] https://viaf.org/
[45] http://www.geonames.org/

archives, libraries or museums, can be mapped to this semantic framework, as it provides a set of compatible models and collaborations, e.g. FRBRoo (a conceptual model for bibliographic information in object-oriented formalism), PRESSoo (a conceptual model for bibliographic information pertaining to serials and other continuing resources), or CRMba (an extension of CIDOC CRM to support buildings' archaeological documentation). Also compliant to CRM are LIDO (Lightweight Information Describing Objects) and EDM (Europeana Data Model). CRM is independent of any encoding syntax and may be used with XML or RDF.[46]

Europeana Data Model (EDM)

Europeana created EDM in 2009/10 in order to get a linked data compliant model for presentation and data exchange in the Europeana portal.[47] Based on concepts and terms from Dublin Core and CIDOC CRM, this model was developed to allow a description of Cultural Heritage Objects that would be not as complex as specified by CRM, but more sophisticated than Dublin Core.[48] Since then, EDM evolved into a de facto standard, being used not only by different Europeana projects but also in applications independent from Europeana funding, like the Digital Public Library of America (DPLA) and the Deutsche Digitale Bibliothek (DDB).

Darwin Core (DwC)

DwC is an ontology for the description of biological diversity and was created and published by the Darwin Core Group, a task group of the Taxonomic Database Working Group (TDWG).[49] Based on Dublin Core and extended by terms defined by DwC[50], the ontology can be used to describe taxon, their occurrences, specimens, samples and related information. The model is syntax independent and may be used with XML and RDF. The DwC Group also developed a mapping to the Access to Biological Collections Data (ABCD) standard in order to support the compliance with legacy data.[51]

Nomisma Ontology

The Nomisma Ontology was developed by nomisma.org, a collaborative project with the purpose to design a thesaurus for numismatic concepts and to provide URIs for those concepts. The Nomisma Ontology enables the linked data compliant description of these concepts and their publication as linked open data.[52]

5.4 Domain specific KOS

Art and Architecture Thesaurus (AAT)

The AAT is a thesaurus that provides unique identifiers for concepts used to describe entities in the fields of art, architecture, anthropology, etc. Hosted and maintained by

---

[46] http://www.cidoc-crm.org/
[47] https://www.europeana.eu/portal/en
[48] https://pro.europeana.eu/resources/standardization-tools/edm-documentation
[49] http://www.tdwg.org/activities/darwincore/
[50] http://rs.tdwg.org/dwc/terms/index.htm#theterms
[51] http://rs.tdwg.org/dwc/terms/history/dwctoabcd/index.htm
[52] http://nomisma.org/

the Getty Research Institute in collaboration with other organizations worldwide, the thesaurus is mainly used in the museum world. Translations of the thesaurus in other languages (e.g. Dutch and Spanish) give a broader access to the concepts and facilitate their usage worldwide. The thesaurus is freely available as Linked Open Data and may be integrated in any application.[53]

Iconclass

Iconclass is a classification describing subjects of images (e.g. paintings, drawings, photographs, etc.). Hosted by the Netherland Institute for Art History (RKD), the thesaurus is freely available as Linked Open Data. Iconclass provides URIs for every notation and allows a multilingual access to the subject headings. Designed for art and iconography, the thesaurus is mostly used in the museums and art institutions around the world.[54]

Encyclopedia of Life (EOL)

EOL is an international organization providing global access to taxonomic and other information about plants, animals, fungi, and microbes.[55] Matching data from resources across the world (mostly museums and learned societies), EOL provides a database that is freely available without any access restrictions. All entities in this database have URIs[56], and most of the content is published under Creative Commons licenses.[57]

MARC Code List for Relators (marcrel)

Marcrel is a list of terms and codes, specifying the relation between agents (persons, organizations, etc.) and other entities. The list is hosted by the Library of Congress and is freely available. Downloads of the complete list are available in different formats (RDF, JSON, MADS, SKOS, etc.). The relators are identified by URIs and defined as RDF properties using the Metadata Authority Description Schema (MADS), and as concepts using SKOS.[58]

LIDO event type vocabulary (eventType)

The eventType vocabulary of the LIDO Terminology is a hierarchical list of terms describing the character of events on a general level. The terminology is hosted by the digiCULT-Verbund and is maintained by the LIDO Terminology Group, a task group of the ICOM. Terms and URIs are published on the xTree.public site.[59] The data is available in skos-thes, an extension of SKOS. Download of the data is possible via SPARQL.

[53] http://www.getty.edu/research/tools/vocabularies/aat/about.html
[54] http://www.iconclass.nl/home
[55] http://www.eol.org/about
[56] http://www.eol.org/info/uniform_resource_identifiers
[57] http://www.eol.org/info/copyright_and_linking
[58] http://id.loc.gov/vocabulary/relators.html
[59] http://terminology.lido-schema.org/eventType

6. Selecting and Defining Metadata Terms

A single ontology, defined by others, often only partially covers the requirements of a specific context. Sometimes there may be a lack of appropriate classes and properties or the description of an entity is not possible in the needed depth. However, we may find out that an appropriate description is possible using classes and properties offered by term sets from other ontologies, i.e., as DCMI calls it, a mixing and matching of terms from different namespaces.

> *"This notion of mixing and matching is familiar to users of XML specifications, who have a long history of defining XML elements per namespace and allowing a document to draw on elements from a variety of namespaces."*[60]

If we wish our data to be used by a broader audience, it is recommended to choose terms from cross-domain models like the Dublin Core Metadata Element Set or the W3C Provenance Ontology. Such standards are capable for a fundamental description of our entities, they are proved and tested in practice, and they are widely recognized by a broader audience. In addition to those widely used terms, we will still have the need for properties and classes matching our specific requirements. Mixing terms from a cross-domain standard like Dublin Core with terms from a domain specific standard like Darwin Core allows a more granular description of our entities while supporting the reuse of the data in a general context. Moreover, if we want to publish our data as linked data, it must be not only human readable but also understandable for machines. Therefore, properties and classes forming the model must be identified by Uniform Resource Identifiers (URI) and interlinked via RDF triple.[61]

Reviewing metadata models that seem appropriate for our purpose does not only mean to compare them with the requirements we developed in our use case analysis[62]. In addition, the following questions should be answered affirmatively:

1. Are the terms common within our community?
   Check, whether other organizations of our domain use the same terms or are familiar with them.
2. Is the maintenance and deployment of the terms guaranteed?
   Check the reliability of the organization(s) responsible for the maintenance and development of the terms.
3. Are the terms freely available?
   Check ownership rights and licenses. Restrictions according to access and use of terms may prevent the reuse of our model and our data by others.

---

[60] Harper, Corey A.: Dublin Core Metadata Initiative: Beyond the Element Set. In: NISO Information Standards Quarterly, 2010, Vol. 22, Issue 1.

[61] Cf. Chapter 2.1.

[62] Cf. Chapter 3.1.

4. Are the selected terms suitable for mixing and matching in the context of Linked Data?
   Check the compliance of the terms with RDF and Linked Data requirements.

In this chapter, we show an example of the mixing and matching of terms from different standards. We also describe how to define application specific terms in a linked data compliant way and show how to align them with terms from metadata standards. At the end of the chapter we give a brief overview of the challenges that the definition of application specific terms entail.

6.1 Mixing and Matching

After the creation of the domain model and the scanning of different standards, we have to decide, which terms we want to use in our application. Therefore, we compare the term definitions of standard terms with the definitions of the properties and classes of our application. If they correspond (e.g. our definition of "Item" with the definition of "Physical Object" in the DCMI Type Vocabulary[63]) or the definition of the standard term is more general but includes our definition (e.g. our property "has name" and the property "title" defined in the DCMI Metadata Terms[64]), it becomes a candidate for the use in our profile. However, to make sure that a term is compliant to our needs, we also have to check whether constraints defined in the standard may contradict the constraints of the property we defined in our example. This is particularly important with respect to domain and range of the properties. For example, we defined our property "has name" as a datatype property, which can be used for the description of Items as well as Events. Because of these requirements, the term we use must be usable with more than one kind of entity, and the range must be a literal.[65] This consequently means that we can't use the property "P102 has title" defined by CIDOC CRM[66], because the domain of this property has to be an instance of the class E 71 Man Made Thing, and in CIDOC CRM an event can't be an instance of this class[67]. Added to this, the range of the property must be an instance of the class "E35 Title", i.e. a URI identifying an entity and not a literal. Therefore, a better choice would be the use of the term "title" defined in the DCMI Metadata Terms.[68] This property has no constraints with respect to the domain, and regarding the range a literal has to be used, as it is the case with our property.

For our example, we decided to use terms from the DCMI vocabularies for the description of instances of the class "Item", and terms from the nomisma.org vocabulary for those properties specific for the "Numismatic Item". For the description of "Concepts", we use the SKOS vocabulary, because we want to link with KOS standards

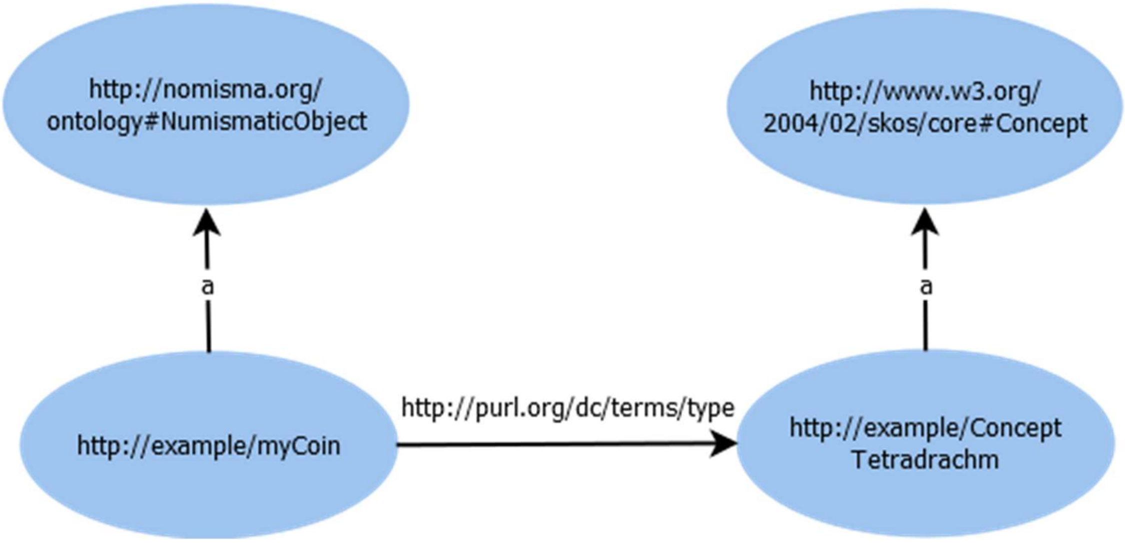
[63] http://dublincore.org/documents/dcmi-terms/#section-7
[64] http://dublincore.org/documents/dcmi-terms/#section-2
[65] Cf. Chapter 4.2.
[66] http://www.cidoc-crm.org/Property/P102-has-title/Version-6.2
[67] http://www.cidoc-crm.org/entity/e71-man-made-thing/version-6.2
[68] http://dublincore.org/documents/dcmi-terms/#section-2

using SKOS. As cross-domain vocabularies the DCMI vocabularies and SKOS support the use of our data in a broader context, and nomisma.org allows the reuse in a numismatic context. All three vocabularies are freely available, and linked data compliant, and maintained by organizations with a reliable background. In the following tables, the entities and properties from our example are organized by their role in an RDF triple as subject, predicate, and object. Below them are listed the terms from the standards we chose, identified by URIs.

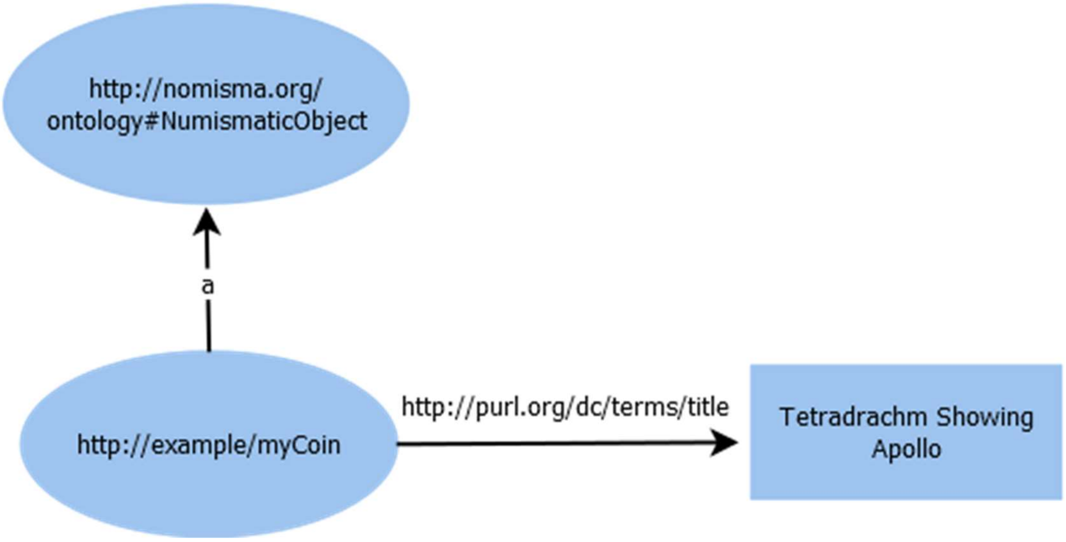
Subject	Predicate	Object
Item	has name	
http://purl.org/dc/terms/PhysicalResource	http://purl.org/dc/terms/title	http://www.w3.org/2000/01/rdf-schema#Literal
Item	has representation	Digital Representation
http://purl.org/dc/terms/PhysicalResource	http://www.cidoc-crm.org/cidoc-crm/P138i_has_representation	http://www.europeana.eu/schemas/edm/WebResource
Item	has type	Concept
http://purl.org/dc/terms/PhysicalResource	http://purl.org/dc/terms/type	http://www.w3.org/2004/02/skos/core#Concept
Numismatic Item	has material	Concept
http://nomisma.org/ontology#NumismaticObject	http://nomisma.org/ontology#hasMaterial	http://www.w3.org/2004/02/skos/core#Concept
Numismatic Item	has depiction	Concept
http://nomisma.org/ontology#NumismaticObject	http://nomisma.org/ontology#hasIconography	http://www.w3.org/2004/02/skos/core#Concept

The following graphs illustrate the matching of terms from different standards in statements about an entity, and the differences between the use of an object property and a datatype property in RDF. For the property “has type” we use the URI http://purl.org/dc/terms/type, an object property. Please note that not only the subject has to be an entity identified by a URI but also the object. In addition, both entities have to be declared as instances of the class they belong to. In the case of the subject this is the class http://nomisma.org/ontology#NumismaticObject, defined by nomisma.org[69], in the case of the object it is the class http://www.w3.org/2004/02/skos/core#Concept, defined by SKOS.

69 http://nomisma.org/ontology.161114.rdf



For the property "has name" we use http://purl.org/dc/terms/title, a datatype property where the subject is the same instance of the same class as in the graph below, but the object is a string, in this case the designation of the coin.



6.2 Defining Metadata Terms

Although it is highly recommended to use already defined terms from metadata standards that would guarantee the reusability, it may be possible that the search for a term fitting our purpose will be unsuccessful. If this is the case, we can define our own property or class. However, if we want to define our own linked data compliant term, we have to respect the rules of semantic contextualization as defined in the Linked Data principles by Tim Berners Lee.[70] This means that our term must be unambiguously defined by

- a URI for the unique identification of the term,
- a machine-readable description of the term, accessible via the URI,

[70] Cf. Chapter 2.1.

- a human readable definition describing what the term is about,
- an alignment to corresponding standard terms (as subclass or sub-property).

The following table shows properties and classes we defined for our example. The decision to create a specific ontology for the description of events was driven by the fact, that the use of properties and classes defined by CIDOC CRM seemed too complex for our example, and the use of the PROV-O seemed too general. Nonetheless, for the description of name, type, and date, we decided to use properties from the DCMI vocabularies. Because the Dublin Core Metadata Element Set[71] does not specify the range of the property "date", this property would be a good choice if our object was a literal (e.g. an ISO 8601 structured date information), or an entity (e.g. a period we want to say more about).

Subject	**Predicate**	**Object**
Item	was affected by	Event
http://purl.org/dc/terms/PhysicalResource	http://example/pdo/wasAffectedBy	http://example/pdo/Event
Event	has name	
http://example/pdo/Event	http://purl.org/dc/terms/title	http://www.w3.org/2000/01/rdf-schema#Literal
Event	has type	Concept
http://example/pdo/Event	http://purl.org/dc/terms/type	http://www.w3.org/2004/02/skos/core#Concept
Event	took place at	Place
http://example/pdo/Event	http://example/pdo/tookPlaceAt	http://example/pdo/Place
Event	occurred at	Time
http://example/pdo/Event	http://purl.org/dc/elements/1.1/date	http://example/pdo/Time or http://www.w3.org/2000/01/rdf-schema#Literal
Event	has participant	Agent
http://example/pdo/Event	http://example/pdo/participant	http://example/pdo/Agent

Good practice is to describe the terms using RDF triple statements. RDF-based ontology languages are the Resource Description Framework Schema (RDFS)[72] and the Web Ontology Language (OWL)[73]. RDFS, a W3C recommendation, is an extension of the basic RDF vocabulary and a simple ontology language used to create a semantic model for RDF data. While RDF defines a syntax for the description of entities, RDFS is used to define the properties and classes used for the description of those entities. OWL, another W3C standard, has a more complex structure than RDFS. It uses some basic terms of RDF and RDFS, but applies additional OWL terms and is more appropriate for computer inferencing. According to their needs, developers have the choice to use one of the

[71] http://dublincore.org/documents/dcmi-terms/#section-3

[72] https://www.w3.org/TR/rdf-schema/

[73] https://www.w3.org/OWL/

three sublanguages of OWL: OWL Lite, OWL DL, and OWL Full. Most ontology developers prefer OWL because of its advantages compared with RDF: the richer vocabulary, and the possibility to make annotations and express restrictions.

There are various tools for designing ontologies and their terms. To get an overview visit the "Ontology editors" list, published by the W3C.[74] If you have to define a complex ontology, we recommend the use of an editor. However, depending on the system and your knowledge, to set up such a tool can become a lengthy process, and the training can be time-consuming. For the definition of a handful of terms, it is therefore often sufficient to use a simple text document. The following example shows the definition of two classes and a property using RDF Turtle[75] as syntax and RDFS and OWL for the description of the terms.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix crm: <http://erlangen-crm.org/170309/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix edm: <http://www.europeana.eu/schemas/edm/> .
@prefix dwc: <http://rs.tdwg.org/dwc/> .
@prefix dcmitype: <http://purl.org/dc/dcmitype/> .
@prefix pdo: <http://example/pdo/> .

pdo:Event a owl:Class ;
  rdfs:label "Event"@en , "Ereignis@de" ;
  rdfs:subClassOf crm:E5_Event , dcmitype:Event , edm:Event , prov:Activity ,
                  dwc:Event ;
  rdfs:isDefinedBy <http://example/pdo/Event> ;
  rdfs:comment "Something that happened and affected the lifecycle of the
  resource."@en .

pdo:tookPlaceAt a owl:ObjectProperty ;
  rdfs:label "took place at"@en , "geschah"@de ;
  rdfs:subPropertyOf dc:relation , edm:happenedAt , prov:atLocation ,
                     crm:P7_took_place_at , dwc:habitat ;
  rdfs:isDefinedBy <http://example/pdo/tookPlaceAt> ;
  rdfs:comment "Relates an event with the place of this event."@en .

pdo:Place a owl:Class ;
  rdfs:label "Place"@en , "Ort"@de ;
```

[74] https://www.w3.org/wiki/Ontology_editors

[75] RDF Turtle is a simple textual syntax for the description of an RDF graph. Cf. https://www.w3.org/TR/turtle/

```
  rdfs:subClassOf dcterms:Location , edm:Place , prov:Location , crm:E53_Place ;
  rdfs:isDefinedBy <http://example/pdo/Place> ;
  rdfs:comment "A geographic location"@en .
```

At the top of the document, we declare the namespaces of the terms we use. The use of a prefix instead of a URI supports the readability of our ontology also for humans. Next, we have to specify the type of term as either a class, an object property, or a datatype property. In our example, two terms are declared as owl:Class and one as owl:ObjectProperty. To align our terms with terms from other ontologies, we declare them as sub terms (rdfs:subClassOf or rdfs:subPropertyOf) of those terms with overlapping semantics that are relevant for our domain. This declaration indicates that the terms we align have a broader meaning, and this way the alignment supports the transformation of our data into schemes more widely used. For the human understanding of the term, we use the label and the comment. The label gives the term a name, and the comment describes what the term is about. In both cases we get text that should be specified by its language (in our example we use English and German).

If we want to support the verifiability of the use of our terms, the domain and range of the properties should be defined.

```
pdo:tookPlaceAt a owl:ObjectProperty ;
  rdfs:label "took place at"@en , "geschah"@de ;
  rdfs:subPropertyOf dc:relation , edm:happenedAt , prov:atLocation ,
                          crm:P7_took_place_at , dwc:habitat ;
  rdfs:domain pdo:Event ;
  rdfs:range pdo:Place ;
  rdfs:isDefinedBy <http://example/pdo/tookPlaceAt> ;
  rdfs:comment "Relates an event with the place of this event."@en .
```

Constraints on domain and range support the validation of data and make sure that the data is compliant to our application. However, if we want our terms to be used in a broader context, it is advisable to define them not too restrictive, because constraints may prevent others from using our properties. As we have seen in chapter 6.1., domain and range that doesn’t fit our purpose results in the decision not to use the CRM property “P102 has title” for the name of an entity, although the semantic of the terms is overlapping. Instead, we use the Dublin Core property “title”, which has no constraints on the domain. If we define domain and range for our property pdo:tookPlaceAt as above, this property can only be used to describe instances of the class pdo:Event. If we want to use an event class from another vocabulary, the data would be invalid. The same applies for the range, where the entity must be an instance of the class pdo:Place. Therefore, we define the properties of our ontology without domain and range. This way we make it easier for others to reuse our terms if they want to describe similar things. For the validation of our data, we decided to design a Description Set Profile where we

mix and match the terms from our ontology with terms from different vocabularies and specify them by the constraints relevant for our application.[76]

6.3 Management of the Terms

> *"… an important design feature of RDF is that metadata vocabularies are easy to define …"*[77]

As we have seen in the last chapter, it is relatively simple to design our own terms and define them in a machine-readable way. We have to be aware, however, that there is also work that has to be done in the long run. Our terms must be retrievable for those who want to use our data, because they need to understand the underlying semantic context of the used vocabularies. Furthermore, we want our terms to be used by other applications, because this supports the interoperability of the data at all. To meet these requirements, the definition of the terms must not only be machine-readable but also

- freely available,
- permanently accessible,
- valid and consistent,
- and all this in the long run.

So before we start defining new terms, we first should resolve how and by whom the terms will be published and maintained. One of the easiest tasks seems to be the publication in the Web. Today, it is simple to provide a website with an ontology and related information. However, we all know that web pages may change their address or disappear completely. The answer to this problem are persistent URLs (PURL) that do not directly point to the resource in the web, but to a resolver that answers with the current address of the resource (e.g. the PURL resolver of the internet archive where we may register our ontology). But this does not solve the problem of the long-term-availability of the terms. At least, there must be someone reporting changes of the URL to the resolver. However, what we really need is an organization in the background ensuring the availability of the documentation and the further development of the ontology. In this context, we recommend to cooperate with an organization already involved in the management and development of standards. Such an organization may be a domain specific body as the International Committee for Documentation of the ICOM[78], or a cross-domain body as the Dublin Core Metadata Initiative[79]. Another issue that requires attention if we want to publish linked data compliant ontologies is the content negotiation. This http technique allows the providing of different documents representing the same ontology via the same URL, but in different formats (e.g. our ontology in RDF/Turtle and RDF/XML). For a detailed description we recommend the

[76] Cf. Chapter 7.

[77] Dunsire, G. et al.: Linked Data Vocabulary Management. In: Information Standards Quarterly, 2012 (Vol.24), Issue 2/3 (http://www.niso.org/niso-io/2012/06/linked-data-vocabulary-management).

[78] http://network.icom.museum/cidoc/

[79] http://dublincore.org/

"Best Practice Recipes for Publishing RDF Vocabularies" published by the W3C.[80] To provide our ontology in different formats is a simple task, because there is a variety of converter to be found that can change RDF data from one format to another. The following example is a conversion from our Turtle example above to RDF/XML done by using the EASYRDF-Converter[81].

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <owl:Class rdf:about="http://example/pdo/Event">
    <rdfs:label xml:lang="en">Event</rdfs:label>
    <rdfs:label>Ereignis@de</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://erlangen-crm.org/170309/E5_Event"/>
    <rdfs:subClassOf rdf:resource="http://purl.org/dc/dcmitype/Event"/>
    <rdfs:subClassOf rdf:resource="http://www.europeana.eu/schemas/edm/Event"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/ns/prov#Activity"/>
    <rdfs:subClassOf rdf:resource="http://rs.tdwg.org/dwc/Event"/>
    <rdfs:isDefinedBy rdf:resource="http://example/pdo/Event"/>
    <rdfs:comment xml:lang="en">Something that happened and affected the lifecycle of the
   resource.</rdfs:comment>
  </owl:Class>

  <owl:ObjectProperty rdf:about="http://example/pdo/tookPlaceAt">
    <rdfs:label xml:lang="en">took place at</rdfs:label>
    <rdfs:label xml:lang="de">geschah</rdfs:label>
    <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/relation"/>
    <rdfs:subPropertyOf rdf:resource="http://www.europeana.eu/schemas/edm/happenedAt"/>
    <rdfs:subPropertyOf rdf:resource="http://www.w3.org/ns/prov#atLocation"/>
    <rdfs:subPropertyOf rdf:resource="http://erlangen-crm.org/170309/P7_took_place_at"/>
    <rdfs:subPropertyOf rdf:resource="http://rs.tdwg.org/dwc/habitat"/>
    <rdfs:isDefinedBy rdf:resource="http://example/pdo/tookPlaceAt"/>
    <rdfs:comment xml:lang="en">Relates an event with the place of this event.</rdfs:comment>
  </owl:ObjectProperty>

  <owl:Class rdf:about="http://example/pdo/Place">
    <rdfs:label xml:lang="en">Place</rdfs:label>
    <rdfs:label xml:lang="de">Ort</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://purl.org/dc/terms/Location"/>
    <rdfs:subClassOf rdf:resource="http://www.europeana.eu/schemas/edm/Place"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/ns/prov#Location"/>
    <rdfs:subClassOf rdf:resource="http://erlangen-crm.org/170309/E53_Place"/>
    <rdfs:isDefinedBy rdf:resource="http://example/pdo/Place"/>
    <rdfs:comment xml:lang="en">A geographic location</rdfs:comment>
  </owl:Class>

</rdf:RDF>
```

[80] https://www.w3.org/TR/swbp-vocab-pub/

[81] http://www.easyrdf.org/converter

One of the most important issues we have to handle when we publish an ontology is the question of versioning. Application developers, using terms from ontologies published and developed by others, need a monitoring of the changes to those terms and the ontology, of which they are part. In addition, for the reuse of metadata using these terms, the version of the terms can become crucial. To proof the compliance of the data and handling it, any validation and further processing needs to be done on the terms version actually used in the data. As Dunsire et al. 2012 wrote, the management and providing of different versions of ontologies could become a task that metadata directories like the Open Metadata Registry[82], the Linked Open Vocabularies (LOV)[83], or BARTOK.org may support in the long run.[84] In the meantime, we recommend to publish each new version of an ontology as one new document identified by a new PURL.

7. Design of the Schema

> *"We would no more want to impose the constraints of a single human resources application suite on FOAF and Dublin Core than we would want to assert that such applications need to consume all ontologically valid permutations of FOAF and Dublin Core entities."*[85]

A schema is a documentation describing the structure of the data. It tells people, how the data looks like or should look like. A schema should be a machine-readable documentation that allows the validation of data by machines while creating or processing the data, and it can support the development of input masks and statistical analysis.

If the schema refers to a given application, it is used to proof to what extent the data is compliant with the requirements this application has. If the application is a cataloguing system, a schema ensures that the data is entered into the system according to the requirements. If we want to collect data from different providers, we use the schema to proof whether the data will fit into our data pool or not.

OWL and RDF allow the validation of data, e.g. by setting domain and range of a property. However, the possibilities are limited, because ontologies, as we defined it above[86], are mainly intended to describe the relations between properties and classes used in a broader context. This means, they are created with an open-world assumption[87]. In this open world – as we already showed[88] – it makes no sense to define

[82] http://metadataregistry.org/

[83] http://lov.okfn.org/dataset/lov/

[84] Cf. Dunsire, G. et al.: Linked Data Vocabulary Management. In: Information Standards Quarterly, 2012 (Vol.24), Issue 2/3 (http://www.niso.org/niso-io/2012/06/linked-data-vocabulary-management).

[85] Boneva, Iovka et al.: Shape Expressions Schemas. 16. Nov 2015 (https://arxiv.org/abs/1510.05555).

[86] Cf. Chapter 2.5.

[87] file:///C:/Users/sruehle/Documents/Arbeit/Shape%20Expressions%20(ShEx)%20Primer.htm#relationship

[88] Cf. Chapter 6.2.

the terms too tight. Therefore, conventional tools that validate whether a property is used at all, or a literal value has a specific syntax, are not available. On the other hand, my application profile describes the structure of my specific application. Such an application creates and processes data in a clearly defined expression of terms from one (or more) ontologies where not only rules about domain and range matter, but further issues have to be considered, e.g. the obligation of a property, the repeatability, which values are allowed or not, etc. In this "closed" world of our application, our schema shall proof whether data is compliant to the requirements of this closed world.

Moreover, in our application profile we are matching and merging terms from different ontologies. If we validate our data based on the ontologies, we would have to examine a variety of ontologies, which only partly correlate with our tight application specific requirements. That's why we create an application specific schema, where we include only those terms we use and refine them with our application specific constraints. [89]

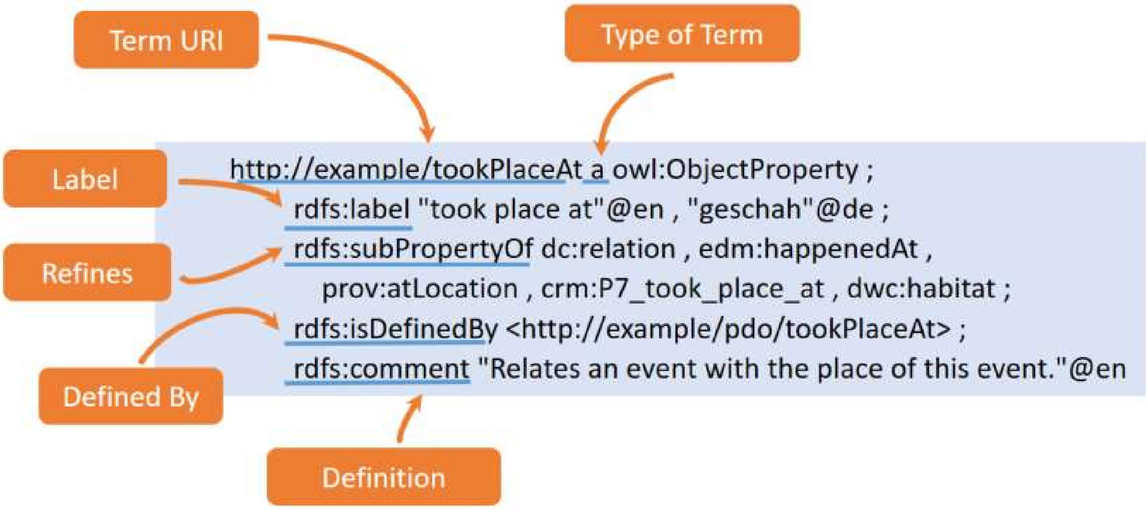
In this chapter, we describe how a schema, created as part of an application profile, may look like. We start with an introduction to the Dublin Core Application Profile Guidelines from 2003 (CWA 14855), which has its origins in the idea of a human readable documentation of metadata structures. In the second section, we describe the concept of the Dublin Core Description Set Profile (DSP), a machine-readable schema that shall also be understandable and usable for non-technical staff. In the last section, we give an overview on the Shape Expressions Language (ShEx), developed by W3C for the validation of RDF data. Another language for the validation of RDF data is the Shapes Constraint Language (SHACL). We will not go into detail concerning SHACL in this guide, a W3C recommendation published in 2017. For those, who want to know more about it, we recommend to read the specification on the web[90] and to try the validator provided there[91].

7.1 CEN Workshop Agreement (CWA) 14855

The „Dublin Core Application Profile Guidelines“, published in November 2003 as CEN Workshop Agreement 14855, is an instruction for a human-readable description of terms in an application profile. It defines the different properties a term has, and specifies which of these attributes are mandatory for the description of a term, and which not. A distinction is made between attributes identifying the terms and those defining it, attributes describing relations between terms, and attributes specifying the constraints. One of the most important statements in these Guidelines is the "Principle of Appropriate Identification"[92], postulating that every term must be identified by a URI. With it, the Guidelines followed Tim Berners-Lees Linked Data principles[93] quite early.

[89] Boneva, Iovka et al.: Shape Expressions Schemas. 16. Nov 2015 (https://arxiv.org/abs/1510.05555).
[90] https://www.w3.org/TR/shacl/
[91] http://shacl.org/playground/
[92] CWA 14855. November 2003 (http://dublincore.org/usage/meetings/2004/03/cwa14855-20040210.pdf), page 6.
[93] Cf. Chapter 2.1.

Though not up-to-date, the guidelines are useful especially for those who are not technical professionals and would have problems with the understanding of a machine-readable schema, let alone that they were able to create one. Furthermore, the guidelines provide an overview of the attributes relevant for the description of a term. Most of the properties used today for the description of terms are already described here, and only the labels differ sometimes. In the following figure, we align the properties we used for the description of terms in our ontology[94] with the CWA 14855 attributes.



According to the guidelines for the description of terms, only the identifying attributes are mandatory. These are the URI, Name, Label and Defined By. For the validation of data that shall fit in our application, however, the constraints are just important. While the attribute "Obligation" specifies whether the use of a term is mandatory or not, the attribute "Occurrence" specifies the repeatability. The attribute "Condition" tells us, in which cases a term must be used (e.g. the obligation to use the term language when describing a text resource, as defined in the EDM Mapping Guidelines by Europeana[95]), and the attribute "Datatype" defines the syntax of the value used with the property. In the following figure, we describe our application specific rules for some of the properties we have chosen in Chapter 6.2. by using the CWA 14855 attributes.

Term URI	http://purl.org/dc/terms/type	http://purl.org/dc/elements/1.1/date	http://example/pdo/tookPlaceAt
Name	type	date	tookPlaceAt
Label	Type	Date	Took place at
Defined By	http://purl.org/dc/terms/	http://purl.org/dc/elements/1.1/	http://example/pdo
Definition	The nature or genre of the resource.	A point or period of time associated with an event	Relates an event with the place of this event.

[94] Cf. Chapter 6.2

[95] Europeana Data Model – Mapping Guidelines v2.4. 06.10.2017. Page 13. (https://pro.europeana.eu/files/Europeana_Professional/Share_your_data/Technical_requirements/EDM_Documentation/EDM_Mapping_Guidelines_v2.4_102017.pdf).

		in the lifecycle of the resource.	
Type of Term	Property	Property	Property
Refines	http://purl.org/dc/elements/1.1/type		http://purl.org/dc/elements/1.1/relation
Has Encoding Scheme	http://terminology.lido-schema.org/eventType		http://www.geonames.org
Obligation	mandatory	mandatory	optional
Condition	There must be one and only one type with a value from the LIDO Eventtype Vocabulary		
Datatype		ISO 8601	
Occurrence	repeatable	not repeatable	not repeatable

7.2 Dublin Core Description Set Profile

Another mandatory part of a Dublin Core Application Profile is the Description Set Profile (DSP). The idea behind the DSP was a schema readable by both, machines and humans. According to that, DSP provides a „simple constraint language" to describe metadata terms used in an application.[96] In addition, the Dublin Core Community tried to knit together the open world assumption of the Linked Data World with the close world assumptions of individual applications. The objective of such a concept was to use, within an application, data from the open world next to data created for this application.

The DSP model assumes that data in an application may describe different sorts of entities (e.g. items and events in our example) related to each other, and that different properties and constraints may be necessary to describe these entities. The Description Set Profile is the place where the classes these entities are instances of are described. Within the Description Set Profile, there is a "Description Template" for every class relevant in the application, specifying what properties we can use with entities of this class, and what are the rules and constraints if we use them. In the "Description Template" the class is set, and it specifies, how often entities, that are instances of this class, may occur in a description. For every property that can be used in this description, a "Statement Template" is created, where a statement is a combination of the property and its value.[97] The "Statement Template" provides the information, what property is used, how often this property must or may be used in the description of an entity, and the character of the values allowed with this property. Dublin Core distinguishes two kinds of values: the "Syntax Encoding Scheme", which is similar to the "Datatype" of the CWA 14855, and the Value Encoding Scheme, specifying a KOS that must be used with a property. But, the use of encoding schemes is not mandatory for the Description Set Profile, as we see in the following example.

[96] http://dublincore.org/documents/singapore-framework/

[97] DCMI Abstract Model. Section 2.2 The DCMI Description Set Model (http://dublincore.org/documents/abstract-model/).

```
DescriptionSet: MyExample

    Description template: Numismatic Item
    minimum = 0; maximun = unlimited
        Statement template: has name
        minimum = 1; maximum = 1
            Property: http://purl.org/dc/terms/title
            Type of Value = "literal"
        Statement template: has type
        minimum = 1; maximun = unlimited
            Property: http://purl.org/dc/terms/type
            Type of Value = "non-literal"
            Value Encoding Scheme URI = http://d-nb.info/gnd/
        Statement template: was affected by
        minimum = 0; maximum = unlimited
            Property: http://example/pdo/wasAffectedBy
            Type of Value = "non-literal"
            defined as = Event

    Description template: Event id=Event
    minimum = 0; maximum = unlimited
        Statement template: has name
        minimum = 1; maximum 1
            Property: http://purl.org/dc/terms/title
            Type of Value = "literal"
        Statement template: has type
        minimum = 1; maximun = unlimited
            Property: http://purl.org/dc/terms/type
            Type of Value = "non-literal"
            Value Encoding Scheme URI = http://d-nb.info/gnd/
        Statement template: occurred at
        minimum = 1; maximum = 1
            Property: http://purl.org/dc/elements/1.1/date
            Type of Value = "literal"
            Syntax Encoding Scheme URI = http://purl.org/dc/terms/W3CDTF
```

In our example, the entities we want to describe are instances of the class Numismatic Item or the class Event. In a description, entities of both classes are allowed, and there may be more than one entity of one of these classes. This is specified by “minimum = 0” (the occurrence is optional) and “maximum = unlimited” (the occurrence is repeatable). The cardinality of the properties is described in the Statement Templates (“minimum = 1” means mandatory, “maximum = 1” means not repeatable), where we also find information about the kind of value allowed in this statement. We distinguish between literal values, i.e. the value is a string of characters, or non-literal values, i.e. the value is a URI representing an entity. As we see, there is no encoding scheme for the description of the Statement Templates “has name”, where we use a simple string, and the Statement Template “was affected by”. The latter is used to describe the relation between the Numismatic Item and the Event, a relation specified by the “defined as” attribute, which points to the Description Template of the Event.

The concept of Description Set Profiles was developed between 2008 and 2009[98] and is deeply rooted in the idea of the record as the keystone of describing objects and their related entities. This way of thinking was always hard to combine with the open world assumption of the Linked Data world, and because of that, the DSP did not really become accepted. However, it is a good practice on our way to a linked data compliant schema, because the DSP embodies most of the information we need for the next step. Furthermore, the language of the DSP doesn't assume any programming knowledge, which makes it easier for non-professionals to understand the structure of the DSP.

7.3 W3C Shape Expressions (ShEx)

ShEx is a schema for the validation of RDF data. In addition, it is also a documentation of the structure of data used for the creation of interfaces und supports the transformation of RDF graphs in other metadata formats. The schema describes the conditions a graph must meet to be compliant with the application for which this schema was created. It defines what triples are allowed in a graph in what combination, what are the cardinalities of the properties, and which datatypes can be used. ShEx has a closed world assumption, because the schema is used to check an already known graph in regard to its reusability for an already known application. The structure of ShEx reminds in many ways of the DSP. For example, the shape of the ShEx Schema can be compared with the Description Set Template, because it describes the triples used for the description and its constraints. As we have already seen above, a triple is always composed of the subject, predicate, and object, where subject and object are the nodes of the triple, and the predicate, represented by a property, is the arc, describing the relation between the two nodes.[99] A shape defines those constraints against which the description of an entity will be checked. The node representing this entity is called the focus node. ShEx differentiates between triple constraints and node constraints. Similar to the Statement Template of the DSP, the triple constraints specify which property is allowed, what the cardinality is, and which kind of nodes are allowed with a specific property. To describe the characteristics of the nodes, node constraints are used.[100] The following example is an implementation of our DSP example from above in ShEx.

[98] http://dublincore.org/documents/dc-dsp/ and http://dublincore.org/documents/profile-guidelines/

[99] Cf. Chapter 2.1.

[100] For a detailed description read the ShEx Primer at http://shex.io/shex-primer/.

```
prefix xsd: <http://www.w3.org/2001/XMLSchema>
prefix dcterms: <http://purl.org/dc/terms/>
prefix pdo: <http://example/pdo/>
prefix dc: <http://purl.org/dc/elements/1.1/>
prefix nmo: <http://nomisma.org/ontology#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>


<NumismaticItemShape> {
    rdf:type [nmo:NumismaticObject] ;
    dcterms:title xsd:string ;
    dcterms:type IRI @<ConceptShape> + ;
    pdo:wasAffectedBy IRI @<EventShape> *
}

<EventShape> {
    rdf:type [pdo:Event] ;
    dcterms:title xsd:string ;
    dcterms:type IRI @<ConceptShape> + ;
    dc:date xsd:date OR IRI
}

<ConceptShape>
    [http://d-nb.info/gnd/.../>~]
AND {
    rdfs:label  xsd:string
}
```

In our example, we use three kinds of entities that we want to describe. It is the above used Numismatic Item, the Event, and the Concept, which we identified as another kind of note relevant for our application. Our focus nodes are therefore instances of the classes nmo:NumismaticObject, and pdo:Event, and entities, where the URI starts with the string http://d-nb.info/gnd/. We created a Shape for each of these classes and listed the properties relevant for them. For the Numismatic Item, these are the properties rdf:type, dcterms:title, dctems:type, and pdo:wasAffectedBy. The default cardinality in ShEx is always one, which means the property is mandatory and not repeatable. If we want to specify another cardinality, we use the regular expression conventions.[101] Because the property dcterms:type is mandatory and repeatable, we use "+" (one or more). The property pdo:wasAffectedBy is optional and repeatable, so we use "*" (zero or more) to specify this. We make further constraints for the nodes. If the entity we describe is a Numismatic Item, the value of rdf:type must be nmo:NumismaticObject, and the value of dcterms:title must be a xsd:string. The constraints for dcterms:type and pdo:wasAffectedBy indicate that the value must be a URI being described according to the Shape that is specified in these constraints. In our examples, these are the EventShape and the ConceptShape.

[101] http://shex.io/shex-primer/#tripleConstraints

The ShEx Compact (ShExC) Syntax we use in our example is based on RDF Turtle, a syntax we also used for the specification of our ontology[102], and is relatively easy to understand. More serialization of ShEx are ShExJ, a JSON-LD based syntax, and ShExR, “the RDF corresponding to ShExJ”. There is also a JSON Format for validation reports.[103]

8. Conclusion

In this Best Practice Guide we describe how to create an ontology and design an application profile. We introduce to key concepts and terms that have to be considered for its design, and that would be compliant to the requirements of interoperability and reusability in a Linked (Open) Data World. Standards help to fulfil this challenging task. The adherence to widely used metadata standards and Knowledge Organization Systems facilitate a consistent definition and description of terms and their relations in a certain knowledge domain that would make them also reusable for other knowledge environments.

Illustrated by an example from practice, we show how to get the information needed to formulate requirements and use cases in order to decide what entities and relations between entities are to select and arrange for ontology and application profile. We show that these entities and properties were mostly selected from different metadata standards, but even missing terms for our contextual purpose could newly be defined. This mixing and making of terms supports interoperability and reusability as long as the terms are conform to the principles of Linked Data.

After selecting the appropriate terms, we show how they can be arranged in order to make certain statements. Depending on the local context, terms are defined as classes, subclasses, or instances of a class, and as object properties, sub-properties, or datatype properties. The decision how to use a term depends on the restrictions made by its definition.

To make an application profile usable, it has to be documented clearly understandable and in a manageable manner. To make it reusable, it has to be published, and to make it reusable for a broader audience, it has to be published under a free license.

[102] Cf. 6.2.

[103] https://www.w3.org/2001/sw/wiki/ShEx#Syntax_and_Serialization_formats